*Journal of Universal Computer Science, vol. 14, no. 16 (2008), 2737-2755*
*submitted: 9/7/07, accepted: 15/10/07, appeared: 28/8/08 © J.UCS*

# Learner Course Recommendation in e-Learning Based on Swarm Intelligence

**Ana-Belén Gil**
(Department of Computer Science, Sciences Faculty
University of Salamanca, Salamanca, Spain
abg@usal.es)

**Francisco J. García-Peñalvo**
(Department of Computer Science, Sciences Faculty
University of Salamanca, Salamanca, Spain
fgarcia@usal.es)

**Abstract:** This paper analyses aspects about the recommendation process in distributed information systems. It extracts similarities and differences between recommendations in e-stores and the recommendations applied to an e-learning environment. It also explains the phenomena of self-organization and cooperative emergence in complex systems coupled with bio-inspired algorithms to improve knowledge discovery and association rules. Finally, the present recommendation is applied to e-learning by proposing recommendation by emergence in a Multi-Agent System architecture.

**Keywords:** Recommendation, E-learning, Multi-agent System, Emergence, Swarm Intelligence
**Categories:** E.4., H.3., H.4., I.2., J.7.

## 1 Introduction

The continued growth and increasing complexity of Web-based applications, from traditional e-commerce, to Web services, to all kinds of dynamic content providers, has led to a proliferation of search tools. Personalized services, such as recommender systems, help engage visitors, turn casual browsers into customers, or help visitors to more effectively locate pertinent information. The goal of any recommendation in any area is to make a selection from among all the possible items by using certain attributes predefined by the context. Customers' preferences toward particular features of products (from books to learning objects) are analysed by different techniques and then rules of customer interest profiles situated within context are derived in order to make the recommendation [Berlanga et al, 2005]. Thus, in general, recommendation processes are defined as the result of several small parameters all together in continuous interaction.

Important characteristics of complex systems have been detected in recommender systems, since emergent behaviour in complex systems can be seen in recommendation systems as well. User behaviour in the decision process is determined not by global control but instead by interactions with the environment. The user has to deal with an extensive amount of partial information. Both systems are thus distributed, large, open, and heterogeneous.

The recommendation process in e-commerce is the task of selecting and organising services according to what the user is interested in. Our approach uses ontological domain knowledge to perform this task. Domain ontologies are valuable in explicit knowledge services representation and in extracting relevant domain semantics to the user profile. The services are semantically defined (e.g. OWL-S [OWL-S, 2006], which provide all the information Web Services need to interact) and thus can be turned into agents semantically labeled with their characteristics. Complex dynamic processes in recommendation turn into simple agent interactions in the environment.

The purpose of this paper is to present a new architecture based on relations between recommenders and complex systems to support recommendation in e-learning working with Learning Objects (LOs). The paper is organized as follows: Section 2 explains a recommendation perspective summarising technologies applied to e-commerce and e-learning. Section 3 presents the characteristics of complex systems and how to attain multi-agent systems implementation with collective behaviour that emerges from a bio-inspired approach. Section 4 introduces the proposed agent-based e-learning architecture. Finally, Section 5 closes the paper, with our conclusions and some ideas for further work.

## 2 The Recommendation Perspective

Before the appearance of Service Oriented Architecture [WSA, 2004], all information resided in static pages on the Internet. Thus, search engines commonly found thousands of potentially relevant sites. For some applications, a user was required to specify his or her goals in terms of a query which was then compared (typically at a simple keyword level) with documents in a collection and those likely to be most related to the query and thus potentially relevant to the user. In the Artificial Intelligence (AI) community, a great deal of work has been done on how AI can help to solve this problem. Notions of personalized search engines, intelligent software agents, and recommender systems gained large acceptance among users for the task of assisting them in searching, sorting, classifying, filtering and sharing the vast amount of information available on the Web. The combination of the modelling of preferences of particular users, building content models, and the modelling of social patterns in intelligent agents provides users with the means for managing information in a rational way and thus helping to overcome information overload.

Nowadays, due to the development of the Semantic Web, we are asked to provide access not just to *static documents* which collect useful information, but also to *services* that provide new ways to offer information for which a new process model must be provided. Service retrieval technology has emerged, but the information retrieval community has focused on the retrieval of documents, not exactly services, and as a result has emphasized a keyword-based approach. As the number of such services increases it will become more and more important to provide tools that allow people (and software) to quickly find the services they need with attention to personalized selection.

## 2.1 Recommenders in E-commerce

These tools in the e-Commerce (EC) environment act as a specialized saleperson for each customer, and they are usually coupled with personalization abilities for each user, based on the analysis of their preferences and interests. Recommenders have mainly relied on user interfaces, techniques of marketing and large amounts of information about other customers and products in order to offer the right item to the right customer [Gil and García, 2006]. Recommenders are the fundamental elements in sustaining site usability and confidence [Egger, 2001], which means they play an important role in the designing of any market place [Spiekermann and Paraschiv, 2002]. EC recommenders are gradually becoming powerful tools for EC business [Gil and García, 2003] covering complex mechanisms mainly in order to support a user's decision process by allowing human beings to use analogical reasoning. Thus, recommenders in EC need to develop in tandem with many areas such as HCI, data-mining, cognitive sciences or marketing. EC sites have made a great effort to supply the customer with tools for making shopping on the Net easier. The need to make sites user-friendly necessarily involves an understanding of consumer behavior in order to facilitate and personalize access to the large amount of information that has to be searched and assimilated before making any purchase.

There are a large number of recommenders with personalization aspects on the Internet. A comprehensive overview of recommenders in EC can be found in [Sarwar et al, 2000], [Shafer et al, 2001], [Montaner et al, 2003]. A rough classification can be made based on the kind of information and the way the recommendation system handles this information to operate. Taking into consideration the purchase system, the consumer's communities or a hybrid of the two as the primary element for building the recommendation, we outline three categories.

1. **Collaborative-social-filtering systems** build the recommendation by the aggregation of consumer preferences. These kinds of systems make matchings to other users based on similarity in behavioural or social patterns. The statistical analysis of data extraction or data mining and knowledge discovery in databases (KDD) techniques (monitoring user behaviour within the system, ratings of the services, purchase history, etc.) build the recommendation by analogies with many other users. Similarities between users are computed using the user-to-user correlation. This technique finds a set of "nearest neighbours" for each user in order to identify similar tastes. Some collaborative filtering systems are Ringo [Shardanand and Maes, 1995] or GroupLens [Konstant et al, 1997]. This technique suffers mainly from the problem of sparsity owing to the need for a large volume of users in relation to the volume of items offered (critical mass) in order to provide the appropriate suggestions. It is also impossible for them to offer new services because there are no prior purchasing records for them. They cannot enter into the dynamic until a large amount of people have previously chosen them.

2. **Content-based-filtering systems** extract the information for the suggestions based on the items the user has purchased in the past. These kinds of systems

use supervised machine learning to induce a classifier to discriminate between services that will be interesting or uninteresting for the user due to her purchasing history. Classifiers can be implemented using many different techniques from artificial intelligence as neural networks, Bayesian networks, inducted rules, decision tree, etc. The user model is represented by the classifier that allows the system to weigh the likes or dislikes for the item. This information identifies the more weighted items that will be recommended to the user. Some content-based systems also use item-to-item correlation in order to identify association rules between items, implementing the co-purchase item. Some examples are [Mooney and Roy, 2000], and Syskill & Webert [Pazzani et al, 1996], where a decision tree is used for classifying web documents attending to some content domain on a binary scale or the well-known recommendation mechanism for the second or third item in Amazon. The above technique suffers mainly from the problem of over-specialization because the consumer is driven to purchase the same kinds of items that they have already purchased. This is a problem also for recommending new articles in the store (no consumer has bought this item before).

3. **Knowledge-based systems** can be understood as a hybrid between collaborative-filtering and content-based systems which also amplifies them. They build knowledge about users linked also to knowledge of services. This information is used to reason what meets the user’s requirements with the item. The relation between services and clients leads to inferences that build the knowledge in the EC engine. Several papers ([Balabanovic et al, 1997], [Shafer et al, 2001], [Hayes et al, 2002]) show the benefits of these systems. Some of these systems provide new solutions for information filtering based on trust.

## 2.2 Technologies in Recommendation

As the problem of recommendation is very widespread, attending to different aspects to solve we can identify two main technological ways to approach it: Web mining and agent-based technologies. Both are also used in additional computer science fields, such as AI.

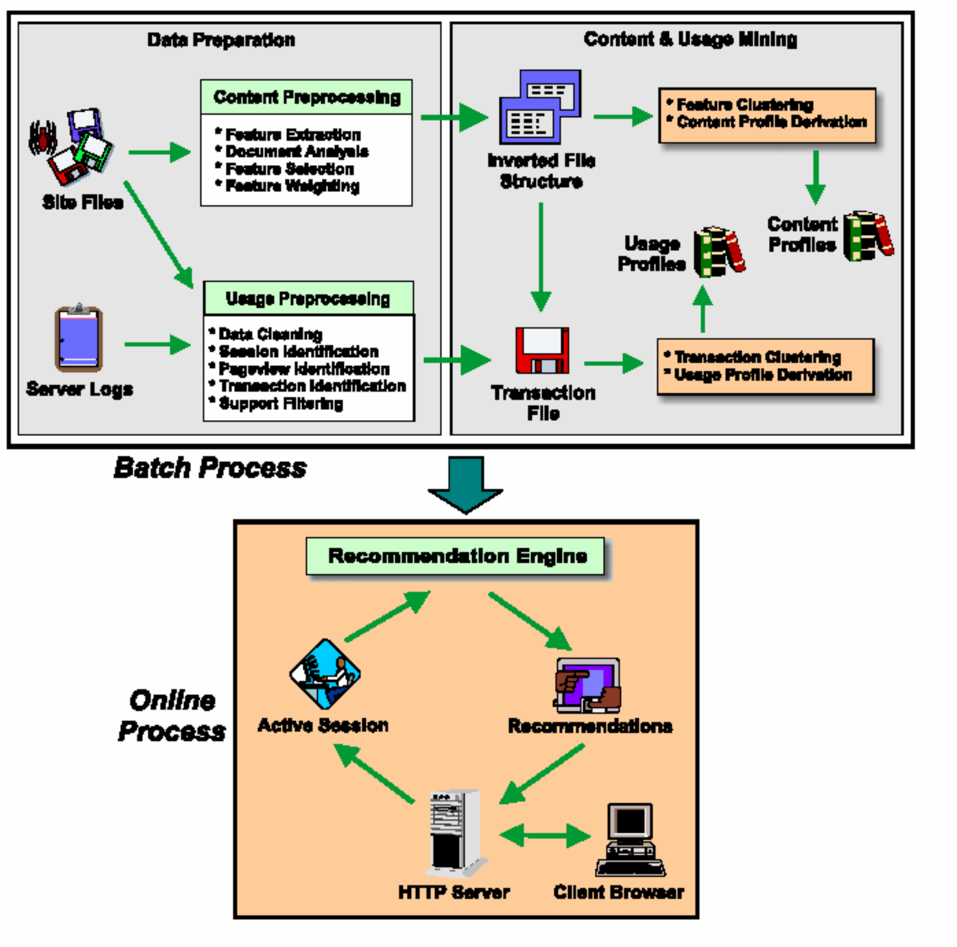

*Figure 1: A general framework for automatic personalization based on Web Mining [Mobasher et al, 2000]*

*Web mining* is the extraction of interesting and useful knowledge and implicit information from artifacts or activity related to the Web. Web servers record and accumulate data about user interactions whenever requests for resources are received. Analyzing the Web access logs can help us understand user behaviour. User profiles are built by combining a user's navigation paths with other data features, such as page viewing time, hyperlink structure, and page content. A comprehensive overview of Web usage mining research (using access logs and mined logs by associating rules and clusters) is found in [Cooley, 2000], [Mobahed et al, 2000] and [Srivastava et al, 2000].

Another path to information retrieval is through agent based applications to filter and present relevant information for the user, which for us is important in our work on these applications based on *ecosystems of adaptive multiagent systems*. A classical view of the agent system mediated architecture is presented in the following figure.

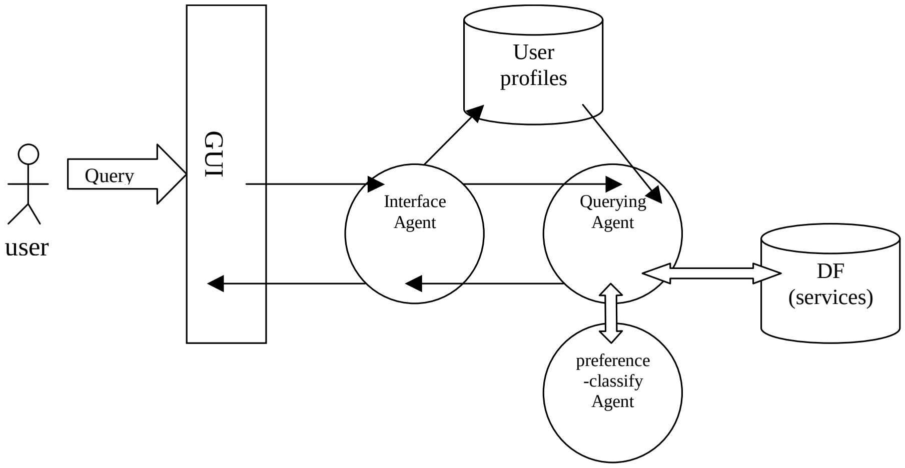

*Figure 2: Agent-based Architecture for the recommender system*

Sheth and Maes, [Sheth and Maes, 1993] implemented an ecosystem architecture of agents to filter Internet News in a system called Newt. A genetic algorithm uses algorithmic analogues to the genetic crossover and mutation operations to generate candidate profiles that inherit useful features from their ancestors, and uses competition to identify and retain the best ones. The crossover operator was periodically applied to combine segments of two candidate profiles which were among those that had produced the highest ranks (using a cosine similarity measure) for articles that the user later identified as desirable. A mutation operator was sometimes applied to the newsgroup name to explore whether existing candidate profiles would perform well on newsgroups with similar names. All of the candidate profiles contributed to the ranking of the documents shown to the user, although those which consistently performed well contributed more strongly to the ranking. Hence, the profile itself was determined by the population of candidate profiles, rather than by any individual candidate.

A similar approach was implemented in Amalthaea [Moukas, 1997] by creating an artificial ecosystem of evolving agents that cooperate and compete in a bounded resource environment. New agents are created by crossover or mutation (or both). Both operators are applied to the evolvable part of the agents, the genotype. The other part of the agents, the phenotype, contains information that should not be evolved, usually instructions on how to handle the evolvable part. The two point crossover operator works as follows: given two agents, it returns two new agents that inherit a part of the keyword vectors of the parents. The operator randomly selects two points in the keyword vector and exchanges all the fields of the two parents that lie between these points, creating two new agents. Mutation is another method for creating offspring agents. The mutation operator takes the genotype of an agent as argument and creates a new agent that is a randomly modified version of its parent. The weights of the mutated keywords are modified randomly while the new mutated keyword is a randomly selected keyword from an agent that belongs to another cluster. The Fab [Balabanovic and Shoham, 1997] and PSUN [Sorensen and McElligot, 1995] systems also implemented this architecture.

However, the size and complexity of the data increases with the syntactical level in recent web advances. Among the most important Web resources are those that provide services. By 'service' we mean Web sites that do not merely provide static information but allow one to carry out some action or change in the world, such as the sale of a product or the control of a physical device. The Semantic Web should enable users to locate, select, employ, compose, and monitor Web-based services automatically.

The generalized term “Web Service” actually does not describe a coherent or necessarily consistent concept; rather, it appears as a new paradigm for the Web by covering the set of technologies, architectures, aspects at different levels of the e-Market or any kind of vision in which software entities are offered with consumer information. It is often used loosely to denote a collection of related technologies, which include: SOAP (Simple Object Access Protocol), Web Services Description Language (WSDL), OWL-S (Ontology Web Language Service), Universal Description, Discovery and Integration (UDDI), etc. The term Web Service determines any perspective in which any software offers information to other software [Gil, 2004].

This means modelling and storing the user model and the content attributes metadata information using standard specifications with metadata structures based on XML. New analysis tools may prove inadequate and more intelligent techniques coupled with metadata treatment are needed in the recommendation process applied to Web Semantic Services recommendation, as summarised in the following figure.

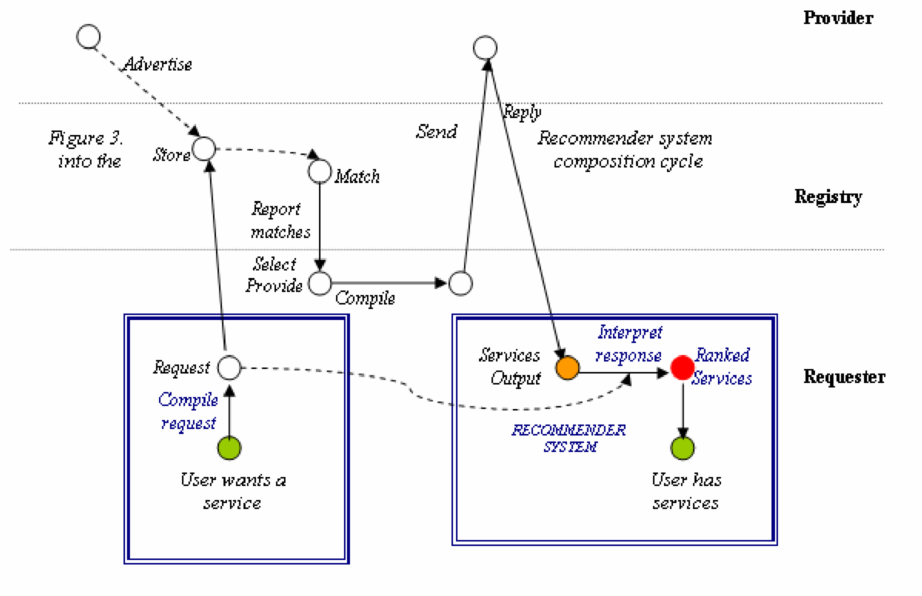

*Figure 3: Recommender system in the composition cycle*

## 2.3 Recommenders in E-learning

The increasing number of e-learning resources means that dynamic educational on-line infrastructures will be needed to manage efficiently all the educational services and elements. E-learning recommendation systems would recommend a learning element to a learner based on the tasks already done by the learner and her successes, and based on tasks made by other 'similar' learners.

The similarity of the learners could be established by employing user profiles, or could be based on common previous access patterns as we explained for e-commerce, but now the context is quite different, because of the appearance of a new role, the tutor. The aim of any learning process is to acquire some contents and the path is stronger than just buying a book.

| Metadata Categories | Metadata Elements |
|---|---|
| 1.General | 1.2 Title<br>1.4 Description<br>1.5 Keywords<br>1.6 Coverage |
| 5.Educational | 5.1 Interactivity Type<br>5.2Learning Resource Type<br>5.3 Interactivity Level<br>5.4 Semantic Density<br>5.6 Context<br>5.7 Typical Age Range<br>5.8 Difficulty<br>5.9 Typical Learning Time,<br>5.10 Description<br>5.11 Language |
| 7.Relation | 7.1 Type<br>7.2 Resource |
| 8.Annotation | 8.3 Description |
| 9.Classification | 9.1 Purpose<br>9.2 Taxon Path<br>9.2.1 Source<br>9.2.2 Taxon<br>9.2.2.1 Id<br>9.2.2.2 Entry<br>9.3 Description<br>9.4 Keyword |

*Table 1: Metadata categories and elements suggested for LO management*

As a consequence of the Semantic Web, an important contribution from computer science to knowledge management and e-learning systems is the learning object (LO) concept. This element has characteristics of independent units, which are able to be reused for other educational situations and platforms. Each of the LOs has metadata

(data about data) for their description and administration. In this way it is possible to know what kind of LO we are dealing with. LOs are characterized by the separation of their content and presentation, and for this reason an important issue to consider is their evaluation and filtering in order to extract a recommedation based on their metadata information. Metadata based on IMS specifications, IMS LOM [IMS LOM, 2006], provide LOs with information on their description and management; thus, it is possible to know whether their characteristics are suitable for other educational situations. The specification includes conformance statements for how meta-data documents must be organized and how applications must behave in order to be considered LOM-conforming.

According to this, knowledge management for e-learning based on reusable units of learning means the possibility to access specific content according to the learners' needs [Morales et al, 2006]. This stage is possible due to standards which were established as an attempt to avoid interoperability platform problems. In order to meet these requirements these elements should satisfy a diverse range of requirements including personalization and adaptation. All these characteristics are well-known in EC recommenders.

In order to support the teaching and learning process through e-learning systems there are a lot of KMS possibilities, such as delivering and evaluating courses, etc. ([Rosenberg, 2001]; [Avgeriou, 2003]). However, according to LOs and standards capabilities, it is necessary to consider how to manage quality LOs, taking into account their characteristics (See Table 1).

We outline the great importance of the recommenders in E-Learning, while we augur a big growth in these tools through the expansion of the Semantic Web

## 3 Multi-Agent System in Complex Systems Implementation

A complex system consists of a large number of interacting units, which when studied in a global perspective could be seen to possess important redundant features and analogies. Furthermore, a complex system is inherently stochastical in its extensive spatiotemporal universe and hence, some of its features could manifest as patterns occurring more frequently than others. These occurrences turn into patterns specifying the repeated aspects of the complex system and the emergence phenomenon appears. In order to describe the relation between the individual dynamic for each unit which becomes a collective dynamic when interacting, we have the term self-organization ([Biebricher et al., 1995], [Bonabeau et al., 1999]). The self-organization phenomenon arises as a spontaneous formation through the evolution and differentiation of complex order structures forming in non-linear dynamic systems by way of feedback mechanisms involving the elements of the systems. But it is by far more crucial that interacting components in complex systems bring about the global structure using only local information, without reference to the global pattern that is forming.

In order to build and study these kinds of complex systems, multi-agent systems (MAS) deal with aspects of cooperation, coalition formation and certain other characteristics that fit with the complex systems description. Each agent has incomplete information or capabilities, no global system control, decentralized data, asynchronous computation and social ability.

Very generally, the elements of the system are treated as multi-agents, relatively autonomous entities which have a set of different rules for interacting with each other. The interaction rules may also be associated with local variables, reducing direct communication among agents which in turn must be strongly influenced by the environmental changes with the necessary flexibility and permeability. By changing the rules of interaction or the influence of the environment during the simulation, one might be able to observe different kinds of collective dynamics and the emergence of new system properties not readily predicted from the basic equations.

Different variations of multi-agent models are applied to simulate socio-economic processes, ecological dynamics, human structure formation, transportation and industrial dynamics, etc., models ranging from ecology to engineering to artificial life. In order to endow the agents in the complex systems with a communication mechanism (this supplies the self-organization) the system is given certain assets based on physical properties ([Parunak et al., 2001] , [Shehory et al., 1999]) either by adapting physics to DAI (Distributed Artificial Intelligence) or by applying organization models extracted from biology [Bonabeau et al., 1999].

### 3.1 Biologic Oriented Agentification

The numerous sorts of social insects and the most well known example, ant colonies, are the inspiration for the organizational models for complex systems [Dorigo and Stützle, 2004]. In general the swarm has to carry out a collective task: each insect deposits a small quantity of chemical substances, called pheromones, allowing it to mark its passage (adding memory) while giving its congeners some information about the environment and information about its own state (communication). Two individuals interact indirectly through the environment. In fact, pheromones lead directly to a specific behaviour in the individual who perceives them; this is called stigmergy. According to Ramos [Ramos and Ajith, 2004], stigmergy could be defined as a typical case of environmental synergy of learning via the environment. Pheromones act as chemical transmitters, allowing ants to communicate among themselves over a short distance. Ants are capable of external storage of information in the environment, achieving memory.

In order to understand collective behaviour, computer simulations are used to examine the parameters and their interactions. In general, these insect colonies in the real world provide three operations on chemical pheromones that support purposive actions in agents' models. They aggregate deposits from individual agents, evaporate them over time (with the effect of avoiding overloading and forgetting obsolete information), and diffuse them to nearby locations (with the effect of providing a gradient that agents can follow). In such an ecosystem we can identify the following agent properties:

- *Autonomous entity*. Each agent acts independently and asynchronously to satisfy its goal. This implies distribution into separate smaller functions.
- *Able to act in its environment.* The basic interaction between the agent and the environment can be considered as indirect chemical communication mediated by an external storage.

- *Knows its environment partially.* This knowledge is based on the interaction of the agent at a local or microscopic level. Each agent works within a bounded rationality. It adapts constantly.
- *Works towards individual goals*.
- *Able to interact with other agents*. The agent has social interactions.
- During the evolution of the system, due to the interaction between agents in the environment, a *collective behaviour emerges by adaptation*. This behaviour is observed on the macroscopic level. Some multi-agent characteristics thus appear:
- *Aggregation* of numerous agents due to *common characteristics* discovered along the evolution in the tasks.
- The multi agent as a system has a goal to achieve.
- No agent controls the global task. Ants perform impressive feats of coordination without direct inter-agent control.
- The environment is dynamic and/or incompletely described.

In a more general view, ecosystems are complex biological systems in which an essential characteristic is adaptation. Some mathematical models of ecosystems simulate models of heterogeneous agents that evolve in a system, according to their fit with some aspect of the ecosystem. Normally these agents compete for resources or work together for a common goal.

## 4 Recommender Systems Proposal for an E-learning Environment

Swarm intelligence uses emergent computing. Dorigo ([Dorigo and Di Caro, 1999], [Dorigo and Stützle,2004]) defines this paradigm with the bio-inspired computational algorithm ACO (Ant Colony Optimization). This algorithm consists of social ants building networks of paths that connect their nests with available food sources. Mathematically, these networks form minimum spanning trees. Thus, they minimize the energy they spend to bring food to the nest. However, general research using artificial ants tends to resolve more difficult issues by adding complexity to the ants' behaviour to solve specific problem domains.

The effort made to extract and study user patterns on the Internet are important, and Ramos [Ramos and Ajith, 2004] proposes an ACLUSTER (Ant Colony Cluster) application to cluster Web usage patterns for predicting the Web traffic volume.

We propose that the recommendation problem can be solved by designing agents covering aspects about learners and LOs. These agents interacting at microscopic level will emerge in a final structure through self-organization from a bottom-top MAS architecture, describing the solution. The management of the elements in the recommendation problem has self-organization aspects that could well be conceived as an ecosystem model because of certain related aspects:

- Similar LO information or related trends to create groups.
- The most needed information in relation to the user's educational needs grows and evolves while the rest is forgetten and disappears.

- The dissemination of all the information in its environment occurs through short-range interaction processes and attends to the diffusion phenomenon by decreasing in time.

Based on the natural selection in the MAS model, each service is represented by an ant while all the ants cooperate and compete to satisfy the personal requirements of the user, who is placed in the environment also as a food source. Both ants and food sources are agents. Only those that describe services sufficiently similar ([Paolucci et al., 2002], [Paolucci et al., 2003]) to the service requested match the request in the degree to which they agree with the profile of the user. These elements form the foundation of the architecture proposed in the next section.

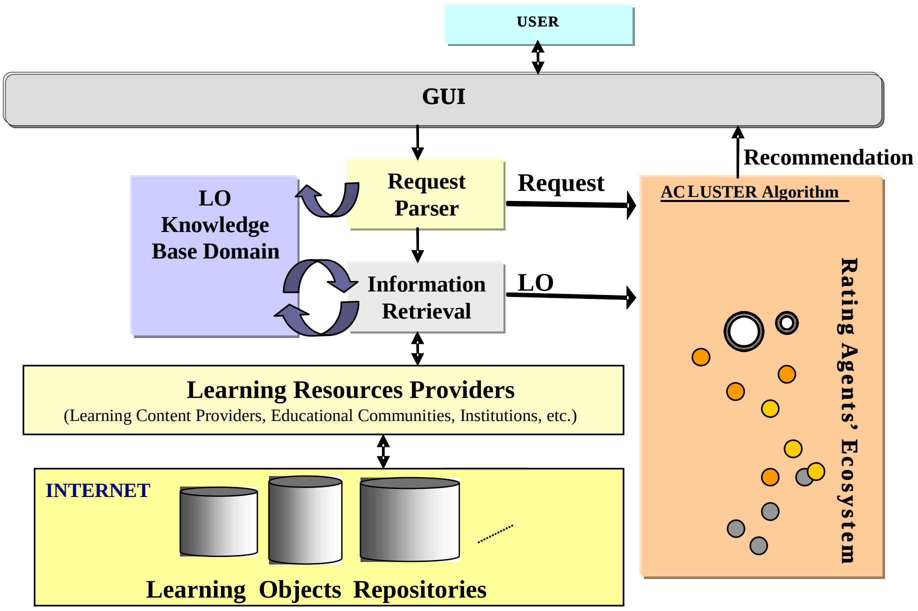

*Figure 4: The recommender system architecture*

### 4.1 Architecture proposal

The system architecture proposed (See Figure 3) is distributed in three layers.

The first layer contains the graphical user interface (GUI) and canalizes the communication data: the user makes her requests, browses, and receives and selects the recommendations. The interaction with the user has strong domain-dependent aspects (described semantically).

The *second layer* contains the main application layer where the recommendation strategies are located . It is conceived as an agents' ecosystem. The ecosystem is the generator of a dynamic representation of the environment in its spatial-temporal evolution. The ecosystem defined in this second layer is composed of a discrete environment where agents are pregnant with user domain characteristics. These agents receive and emit information through the environment.

Finally, the *third layer*, related to the repositories, contains the services knowledge base domain (ontology knowledge) and several extensions to the World Wide Web containing data facilitators. At this level a repository of information for the system is also stored.

### 4.2 The Clustering and Sorting Ant Algorithm

Some kinds of ants, *Pheidole pallidula* [Deneubourg et al, 1991], *Laius niger*[Chretien, 1996] and *Messor sancta* form piles of items such as dead bodies (corpses), larvae, or grains of sand. These ants deposit items at initially random locations. When other ants perceive deposited items, they are stimulated to deposit items next to them, this type of cemetery being a clustering organization, and brood sorting, a type of self-organization and adaptive behaviour. The clustering and sorting behaviour of ants has stimulated researchers to design new algorithms for data analysis. Objects placed next to each other by the sorting algorithm have similar attributes.

Deneubourg et al. [Deneubourg et al, 1991] proposed a first model (called BM, for basic model) where a population of ant-like agents randomly moving on a 2D grid is allowed to move basic objects, piling up those of the same type and building clusters [Chrétien, 1996]. This algorithm was used in the implementation of robots. The probability $p_p$ that a randomly moving, unladen agent (representing an ant in the model) will pick up an item is given by Equation 1, where $f$ is the perceived fraction of items in the neighbourhood of the agent, and $k_1$ is constant. In the same way the probability $p_d$ that a randomly moving loaded agent will deposit an item is given in Equation 2, where $k_2$ is constant.

$$p_p = \left( \frac{k_1}{k_1 + f} \right)^2 \text{ Where } \begin{matrix} if \;\; f << k_1 & p_p \rightarrow 1 \\ if \;\; f >> k_1 & p_p \rightarrow 0 \end{matrix}$$

*Equation 1: BM probability of picking up an item*

$$p_d = \left( \frac{f}{k_2 + f} \right)^2$$

*Equation 2: BM probability of depositing an item*

This method was then further generalized by Lumer and Faieta (hereafter the LF algorithm) [Lumer and Faieta, 1995], who applied it to exploratory data analysis. They showed that their model provides a way of exploring complex information spaces, such as document or relational databases. LF defines a distance or dissimilarity between objects in the space of object attributes. The objects can be described by a finite number of value-related attributes by allowing information access and compared from an n-dimensional space, hence being able obtain different clusters accordingly. The LF algorithm works as follows. Let $d(o_i, o_j)$ be the distance between two objects in the space of attributes. Let us assume that an agent is located at site r at time t and finds an object $o_i$ at that site. The local density $f(o_i)$ with respect to object $o_i$ at a site r is given by:

$$f(o_i) = \begin{cases} \frac{1}{s^2} \sum_{o_j \in Neigh_{(SXS)}(r)} \left[ 1 - \frac{d(o_i, o_j)}{\alpha} \right] & if\ f > 0 \\ 0 & otherwise \end{cases}$$

*Equation 3: LF Local density with respect to object $o_i$ at site r*

$f(o_i)$ is a measure of the average similarity of object $o_i$ with the other objects $o_j$ present in the neighbourhood of $o_i$. $\alpha$ is a factor that defines the scale for dissimilarity. This factor acts as regulator of the similarity allowed between different items to form the same cluster or not. Lumen and Faieta define picking up and dropping probabilities (Equation 4, Equation 5) where $k_1$ and $k_2$ are two constants similar to the ones in the BM.

$$p_p(o_i) = \left( \frac{k_1}{k_1 + f(o_i)} \right)^2$$

*Equation 4: LF picking up probability*

$$p_d(o_i) = \begin{cases} 2f(o_i), & if\ f(o_i) < k_2 \\ 1, & if\ f(o_i) \geq k_2 s \end{cases}$$

*Equation 5: LF dropping probability*

Lumen and Faieta applied the algorithm to a database containing the profile of 1650 bank customers. Attributes of the profiles included marital status, gender, residential status, age, a list of banking services used by the customer, etc. Given the variety of attributes, some of them qualitative, others quantitative, they have to define several dissimilarity measures and combine them into a global dissimilarity measure. Lumer and Faieta evolve the model [Lumer and Faieta, 1994] with some features in order to correct the tendency to create more clusters than those desired. These three new features endow the agents with different moving speeds, together with short-term memory (then agents can remember the last m items they have dropped) and the system is equipped with behavioural switchers that activate some actions or have the possibility to destroy clusters. Despite interesting results, it is not obvious that this algorithm has a future in terms of efficiency in computation time.

Recently, Ramos et al., in different papers [Ramos et al., 2002] [Ramos and Merelo, 2002] and Ramos and Ajith [Ramos and Ajith, 2004] have extended the model by Deneubourg, Lumen and Faieta explained above. This algorithm, called Ant Clustering Algorithm (ACLUSTER), avoids the additional complexity of predecessor algorithms (short-term memory, multiple ant types that move at different speeds) by introducing pheromone trails to achieve unsupervised clustering.

Ramos and Merelo used a redefined Chialvo and Millonas [Chialvo and Millonas, 1995] model wherein an individual ant can be described by its position and direction. The probabilities of an ant moving between any particular pair of position and direction to any other pair are determined by their pheromone weighting function.

$$W(\sigma) = \left(1 + \frac{\sigma}{1+\delta\sigma}\right)^{\beta}$$

*Equation 6: Probability of moving to a location with pheromone density σ(r)*

Where the value $\beta$ determines the randomness with which an ant follows a pheromone trail, $1/\delta$ is the ant's sensory capacity, which describes the fact that each ant's ability to sense pheromone decreases somewhat at high concentration. The normalised probability of going from location i to location k is given by:

$$P_{ik} = \frac{w(\sigma_i)w(\Delta_i)}{\sum_{j/k} w(\sigma_j)w(\Delta_j)}$$

*Equation 7: probability of going from cell i to cell k*

Where $\Delta_j$ measures the magnitude of the difference in orientation from the previous direction at time t-1. Each individual leaves a constant amount $\eta$ of pheromone at the cell in which it is located at every step t, and also this pheromone decays at each time step at a rate k.

The two major factors that influence any local ant action are the number of objects in their neighbourhood and their similarity. Ramos and Ajith [Ramos and Ajith, 2004] defined the probability function for picking up or dropping as a function of different stimulus intensities (number of items and their similarity), at site r:

$$P_p = (1-\chi)\cdot\varepsilon \qquad\qquad P_d = \chi\cdot\delta$$

*Equation 8: Probability functions for picking up or dropping*

Where $\chi$ is defined as the response threshold associated with the number of items, n, present in a 3x3 region around **r,** and d is the similarity between objects as Euclidean normalized distance computed within all the pairs of objects present in the 3x3 region around r.

$$\chi = \frac{n^2}{n^2 + 5^2}$$

*Equation 9: Response threshold associated with number of items in the neighbourhood*

Where $\delta$ and $\varepsilon$ are defined as the response threshold functions associated with the similarity of objects in cases where one is dropped and then picked up later:

$$\delta = \left( \frac{k_1}{k_1 + d} \right)^2 \quad \ldots\ldots\ldots\ldots \quad \varepsilon = \left( \frac{d}{d + k_2} \right)^2$$

*Equation 10: response threshold functions*

ACLUSTER has been applied to clustering textual documents [Ramos and Merelo, 2002], introducing [Ramos and Ajith, 2004] a new type of Data-Mining based on Stigmergic paradigms by hybridizing bio-inspired Swarm Intelligence with Evolutionary computation. This work provides encouraging results.

## 5 Conclusions

This study has taken a brief look at self-organization and emergence while examining certain aspects connected with recommender systems. It explains the foundation from which we propose a recommendation mechanism in e-learning based on complex systems with a bio-inspired algorithm. The environment contains the knowledge of the whole phenomenon, the real facts pertaining to the domain, the user preferences representations and also the characteristics of the LOs, modelled and specified using the IMS Learning Information or LOM specification. By interacting, these elements produce the final representation at macro-level, which customizes a dynamic and personalized representation for the recommendation.

The goal is to have a structural and immediate connection between the set of LO elements that a user needs, relying on the semantic content and qualities evaluation and the landscape of agents that represent the recommendation. We are working on the ACLUSTER algorithm to introduce new metrics that will allow us to identify the similarity between LOs in the environment. The possibility of this proposal is at simulation phase.

This work has attempted to build solutions for real-life applications, based on swarm intelligence, which has great promise for further advancement in this area.

### Acknowledgements

We wish to thank to the GRIAL group (Research Group in Interaction and eLearning) of the University of Salamanca for its contributions and ideas for the development of this work.

This work was partly financed by Ministry of Education and Science as well as FEDER Keops project (TSI2005-00960).

## References

[Avgeriou, 2003] Avgeriou, P., Retails, S., & Skordalakis, M. (2003). An architecture for open learning management systems. In Y. Manolopoulos, S. Evripidou, A.C. Kakas (Eds.) Advances in Informatics. 8th Panhellenis Conference on Informatics. Lectures Notes in Computer Science. LNCS 2563. Berlin: Springer Verlag, pp.183-200.

[Berlanga et al, 2005] Berlanga, A., García, F., Gil, A. "Adaptive Hypermedia Systems: Personalized Education in E-Learning". Educational Virtual Spaces in Practice, The Odiseame Approach, F.J. García, J. García, M. López, R. López and E. Verdú (Eds.), pp. 133-146, Editorial Ariel. ISBN: 84-344-2659-5, 2005.

[Biebricher et al., 1995] Biebricher, C.K., Nicolis, G., Schuster, P., 1995: Self-Organization in the Physico- Chemical and Life Sciences, EU Report 16546.

[Bonabeau et al., 1999] Bonabeau E., M. Dorigo & T. Theraulaz, 1999. From Natural to Artificial Swarm Intelligence. New York: Oxford University Press.

[Chialvo and Millonas, 1995] Chialvo, D.R. and Millonas M.M. How Swarms Build Cognitive Maps. The Biology of Intelligent Autonomous Agents, 144, NATO ASI Series, pp. 439-450. 1995

[Chrétien, 1996] Chrétien, L. Organisation Spatiale du Matériel Provenant de l'excavation du nid chez Messor Barbarus et des Cadavres d'ouvrières chez Lasius niger (Hymenopterae: Formicidae) .PhD thesis, Université Libre de Bruxelles, 1996.

[Deneubourg et al, 1991] Denebourg J.-L., S. Goss, N. Franks, A. Sendova-Franks, C. Detrain, and L. Chretien. The Dynamic of Collective Sorting Robot-like Ants and Ant-like Robots. Procs. of 1st Conf. on Simulation of Adaptive Behavior: From Animal to Animats, edited by J.A. Meyer and S.W. Wilson, pp. 356-365. Cambridge, MA: MIT Press. 1991

[Dorigo and Di Caro, 1999] Dorigo M. and G. Di Caro, 1999. The Ant Colony Optimization Meta-Heuristic. In D. Corne, M. Dorigo and F. Glover, editors, New Ideas in Optimization, McGraw-Hill, 11-32.

[Dorigo and Stützle, 2004] Dorigo, M., Stützle, T. (2004) Ant Colony Optimization. MIT Press.

[Egger, 2001] Egger F. N. (2001). Affective Design of E-Commerce User Interfaces: How to Maximise Percived Trustworthiness. In Proceedings of The International Conference on Affective Human Factors Design. Asean Academic Press, London.

[Gil and García, 2003] Gil, A. and F. Garcia (2003). "E-commerce Recommenders: Powerful Tools for E-business." Crossroads - The ACM Student Magazine 10(2): 24-28.

[Gil and García, 2006] Gil A.B., García, F.J. (2006) "Recommender Systems in E-Commerce". In Encyclopedia of Human Computer Interaction. Idea Group Publishers. Claude Ghaoui Editor. ISBN:1-59140-562-9, 2006.

[Gil, 2004] Gil, A.B. "E-Commerce y Web-Services: Un Paso Adelante en la Interoperabilidad". In Tendencias en el Desarrollo de Aplicaciones Web. F. García Peñalvo y Mª Moreno García (Eds.), Departamento de Informática y Automática de la Universidad de Salamanca, pp. 73-84, 2004.

[IMS LOM, 2006] IMS LOM. (2006). Learning Resource Metadata Specification. http://www.imsglobal.org/metadata/index.html#version1.3

[Konstant et al, 1997] Konstant, J.; Miller, B.; Maltz, D.; Herlocker, J.; Gordon, L.; Riedl, J. Grouplens: applying collaborative filtering to Usenet news. Communications of the ACM, 40(3): 77-87, 1997

[Lumer and Faieta, 1994] Lumer, E. and B. Faieta. Lumer E.D. and Faieta B. Diversity and Adaptation in Populations of Clustering Ants. Procs. of SAB´94 – 3rd Conf. on Simulation of Adaptive Behavior: From Animal to Animats, The MIT Press/Bradford Books, Cambridge, MA. 1994

[Lumer and Faieta, 1995] Lumer, E. and B. Faieta. Exploratory Database Analysis via Self-Organization. Unpublished manuscript, 1995. Referred to in [Bonabeau et al, 1999]

[Montaner et al, 2003] Montaner, M., López, B., de la Rosa, J. Ll., “A Taxonomy of Recommender Agents on the Internet”, Artificial Intelligence Review, Kluwer Academic Publishers. Volume 19, Issue 4, pp. 285-330. June, 2003

[Mooney and Roy, 2000] Mooney, R. J.; Roy, L., Content-Based Book Recommending Using Learning for Text Categorization, Proceedings of the V ACM Conference on Digital Libraries, San Antonio, USA, pp.195-204, 2000

[Morales et al, 2006] Morales, E., García, F., & Barrón, Á. (2006). “Quality Learning Objects Management: A proposal for e-learning Systems” 8th International Conference on Enterprise Information Systems (ICEIS’06). Paphos, Cyprus, 23 - 27 May, 2006.

[OWL-S, 2006] OWL-S 1.2 Pre-Release, March 2006. http://www.ai.sri.com/daml/services/owl-s/1.2/

[Paolucci et al., 2002] Paolucci, M., Kawmura, T., Payne, T., and Sycara, K., 2002. Semantic matching of web services capabilities. In First International Semantic Web Conference.

[Paolucci et al., 2003] Paolucci, M., Sycara, K., and Kawmura, T. Delivering Semantic Web Services, 2003. Tech. Report CMU-RI-TR-02-32, Robotics Institute, Carnegie Mellon University.

[Parunak et al., 2001] Parunak, H. Van Dyke, Brueckner, Sven, 2001. Entropy and Self-Organization in Multi-Agent Systems. Proceedings of the International Conference on Autonomous Agents (Agents 2001), 124-130.

[Ramos and Ajith, 2004] Ramos, V., Ajith Abraham, 2004. Evolving a Stigmergic Self-Organized Data-Mining. In ISDA-04, 4th Int. Conf. on Intelligent Systems, Design and Applications, Budapest, Hungary, ISBN 963-7154-30-2, pp. 725-730, August 26-28.

[Ramos and Merelo, 2002] Ramos, V., Merelo J. Self-Organized Stigmergic Document Maps: Environment as a Mechanism for Context Learning. Procs. Of AEB'02- 1st Spanish Conference on Evolutionary and Bio-inspired Algorithms, Mérida., pp. 284-293. 2002.

[Rosenberg, 2001] Rosenberg, M.J. (2001). E-learning. Strategies for delivering knowledge in the digital age, Mc Graw Hill.

[Sarwar et al, 2000] Sarwar, B.; Karypis, G.; Konstan, J.; Riedl, J. Analysis of Recommendation Algorithms for E-Commerce. ACM Conference on Electronic Commerce, pp.158-167, 2000

[Shafer et al, 2001] Shafer, J.; Konstan, J. A.; Riedl, J. E-Commerce Recommendation Applications. Data Mining and Knowledge Discovery, 5 (1/2), pp. 115-153, 2001

[Shardanand and Maes, 1995] Shardanand, U.; Maes, P. Social Information Filtering: Algorithm for Automating "Word of Mouth". Proceedings of ACM CHI'95 Conference on Human Factors in Computing Systems, pp. 210-217, 1995

[Shehory et al., 1999] Shehory, O, Kraus, S. Yadgar, O., 1999. Emergent cooperative goal-satisfaction in large scale automated-agent system. Artificial Intelligence Journal Vol 110(1).

[Spiekermann and Paraschiv, 02] Spiekermann, S.; Paraschiv, C. (2002). Motivating Human–Agent Interaction: Transferring Insights from Behavioral Marketing to Interface Design. Electronic Commerce Research, Vol. 2, pp. 255–285.

[WSA, 2004] Web Service Architecture. W3C Working Draft 11 February 2004. http://www.w3.org/TR/ws-arch/